

TP MatLab

Pour pouvoir lancer cette préparation de TP, il faut run tp.m

1. Question 1 :

```
1      % fonction pour faire une impulsion de Dirac
2  [-] function[x1]=impulsion(N)
3  [-]     x1=zeros(N,1); % on cree une matrice N * 1 de Zeros
4  [-]     x1(N)=1; % on affecte l'impulsion de dirac à la valeur N demandée
5  [-]     % affichage du graphique :
6  [-]     % on affiche x1 pour les valeurs allant de 0 à la longueur de x1 -1
7  [-]     plot(x1);
8  [-]     xlabel('Temps en seconde'); % legende en abscisse
9  [-]     ylabel('Amplitude'); % legende en ordonnée
10 [-]    title('Impulsion de Dirac'); % titre du graphique
11 [-]    end
```

Screenshot 1 : contenu du script impulsion.m

2. Question 2 :

```
1      % fonction pour creer un echelon d'Heaviside
2  [-] function[x2]=echelon(N)
3  [-]     x2=zeros(N,1); % on cree une matrice de zeros 100 *1
4  [-]     % le but de cette boucle for est de faire demarrer notre boucle à notre
5  [-]     % instant N voulu
6  [-]     for k=N:length(x2) % on fait une boucle qui va de N à la longueur de x2
7  [-]         x2(k)=1; % on affecte la valeur 1 à toutes les valeurs de la matrice x2
8  [-]     end
9  [-]     % affichage du graphique
10 [-]    % on affiche x2 pour les valeurs allant de 0 à la longueur de x2-1
11 [-]    plot(x2);
12 [-]    xlabel('Temps en seconde');
13 [-]    ylabel('Amplitude');
14 [-]    title('Echelon');
15 [-]
16 [-]    end
```

Screenshot 2 : contenu du script echelon.m

3. Question 3 :

```
1  [-] function[x3]=porte(N,theta)
2  [-]     x3 = ones(N, 1); % on cree une matrice de longueur N avec que des 1
3  [-]     x3(theta:N)=0; % pour toutes les valeurs de theta à N, on affecte 0
4  [-]     plot(x3); % on affiche x3
5  [-]     grid on; % on affiche les grilles
6  [-]     % on applique les legendes
7  [-]     title("fonction porte");
8  [-]     xlabel("Temps en seconde");
9  [-]     ylabel("Amplitude");
10 [-]    end
```

Screenshot 3 : contenu du script porte.m

4. Question 4 :

```

1 function[x4]=cosinus(N,A,v0,phi)
2 % affichage de la fonction :
3 % pour toutes les valeurs de x4, on calcule la valeur de la fonction et on
4 % le place sur le graphique
5 plot(x4,A*cos(2*pi*v0*x4+phi));
6 grid on; % on affiche les grilles
7 % on affiche les legendes
8 xlabel('Temps en seconde');
9 ylabel('Amplitude');
10 title('Cosinus');
11 end

```

Screenshot 4 : contenu du script cosinus.m

5. Question 5 :

Pour afficher les fonctions, j'ai préféré pour une question de propreté utiliser la fonction subplot qui me permet d'afficher mes 4 fonctions dans une seule figure, voici comment tp.m est codé :

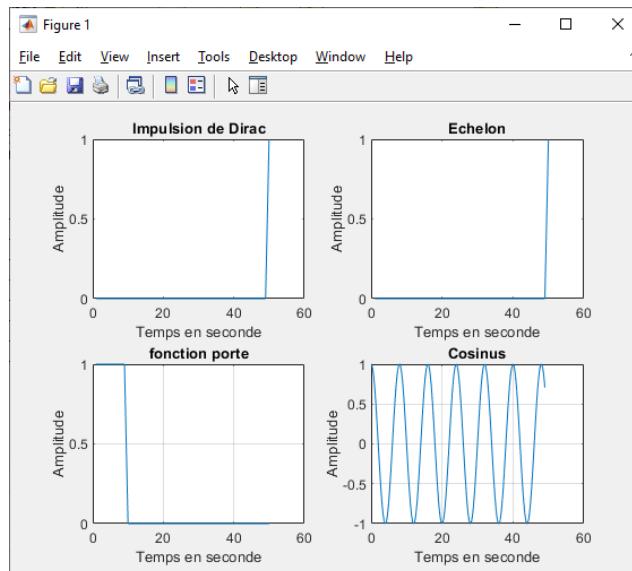
```

1 % la commande subplot nous sert à afficher une fonction dans un cadran
2 % désiré
3 subplot(221);
4 x1 = impulsion(50);
5 subplot(222);
6 x2 = echelon(50);
7 subplot(223);
8 x3 = porte(50,10);
9 subplot(224);
10 x4 = cosinus(50, 1, 0.125, 0);

```

Screenshot 5 : contenu du script tp.m

Ce qui donne :



Screenshot 6 : résultat de l'exécution de tp.m